درس معماری کامپیوتر

استاد: دکتر حمید سربازی آزاد

نیمسال بهار ۱۴۰۱

فاز اول پروژه

اعضای گروه:

پویا یوسفی (۹۸۱۷۱۲۲۳)

محمدرضا احمدی تشنیزی (۹۸۱۷۰۶۴۶)

سجاد پاکسیما (۹۸۱۰۶۲۸۶)

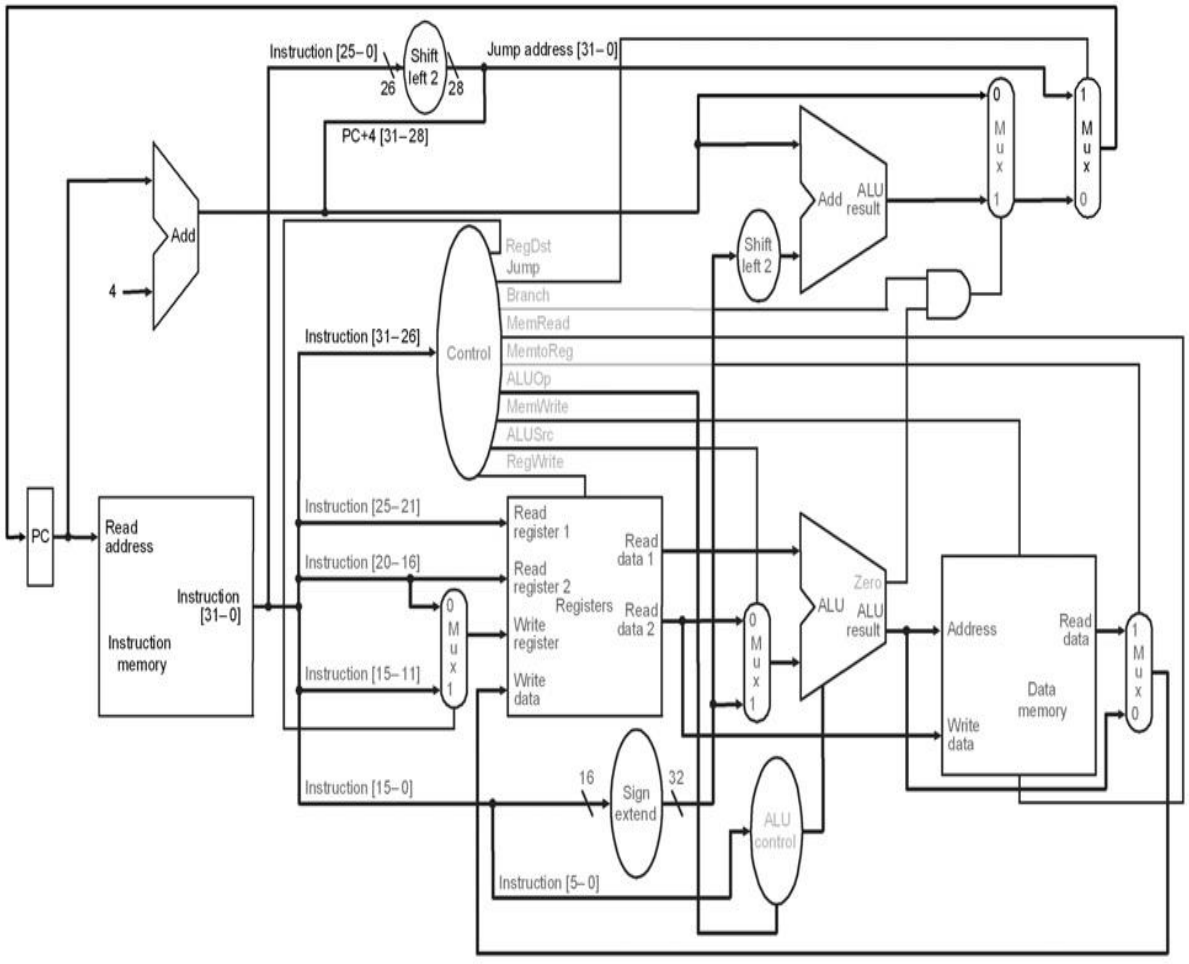
مهدیه ابراهیم‌پور (۹۸۱۷۰۶۲۴)

مقدمه

در این فاز از پروژه، قصد داریم پردازنده‌ای را با استفاده از زبان Verilog طراحی کنیم که بر مبنای معماری MIPS که یک پردازنده RISC است، طراحی کنیم.
پردازشگر MIPS از نوع single-cycle است و وظایف fetch کردن، decode کردن، دسترسی به حافظه و بازنویسی و... را در یک clock انجام می دهد و پاسخ را حاضر می‌کند. در این پردازنده ابتدا مقدار PC به عنوان آدرسی برای شماره‌گذاری دستورات حافظه استفاده می‌شود و مقدار ۳۲بیتی دستور بعدی که باید اجرا شود را به‌عنوان خروجی آماده می‌کند.

مسیر داده (Datapath)

بر اساس مجموعه دستورالعمل‌های ارائه شده، مسیر داده و واحد کنترل به شکل زیر طراحی و پیاده‌سازی شده‌است.



ماژول‌ها:

ماژول Control Unit:

در این ماژول پس از دریافت opdcode و مقدار func، سیگنال های کنترلی یعنی reg_dst, alu_src, alu_opcode و... را مشخص می‌کند. اگر opcode=000000 آنگاه برای مشخص شدن دستور، مقدار func در نظر گرفته می‌شود در غیر اینصورت opcode به تنهایی می‌تواند مقدار سیگنال ها را مشخص کند. برای تشخیص مقدار opcode از دستور case استفاده می‌کنیم. مقادیر ورودی یعنی opcode و func هر دو از دستور گرفته می‌شوند.

ماژول Alu Control:

پس از مشخص شدن سیگنال کنترلی alu_op، در این ماژول عملیات alu را مشخص می‌کنیم. علاوه بر alu_op به ورودی func برای مشخص کردن دستورهای فرمت R نیاز است. مانند ماژول Control Unit، برای تمایز بین دستورات از case کمک می‌گیریم. سیگنال ورودی alu_op از ماژول Control Unit گرفته می‌شود و ورودی func از دستور.

ماژول ALU:

این ماژول، واحد alu است که عملیات‌های مورد نیاز را بر روی ورودی‌های خود انجام می‌دهد. این ماژول به کمک func، sh.amount و alu_select دستورهای R را انجام می‌دهد و برای بقیه دستور ها به کمک alu_select این کار انجام می پذیرد. ورودی‌های sh.amount و func از دستور گرفته می‌شوند و ورودی alu_select از Alu Control.

ماژول Mips Core:

بخش اصلی در این ماژول است که ورودی ها و خروجی ها به کمک سیم‌ها به قسمت‌های مورد نیاز وصل می‌شوند و ارتباط را برقرار می‌کنند. همچنین به کمک مالتی‌پلکسر، می‌توانیم بسته به سیگنال های کنترلی، اتصالات سیم‌ها را به صورت مناسب انجام دهیم. مقدار بعدی PC و مشخص شدن آدرس دستور نیز در این ماژول انجام می‌شود. مقدار فعلی PC بعد از عبور از ادرهای مختلف و مالتی‌پلکسرهای گوناگون، آماده fetch کردن می‌شود. مقدار اولیه PC را برابر با صفر می‌گذاریم چون پردازنده دستورات را از اول برنامه (خط اول) می‌خواند. به کمک سیگنال‌های کنترلی branch, jr, jal, jump پردازنده آماده دستورات پرش می‌شود (قسمت بالایی DataPath). البته برای دستور branch به سیگنال zero که خروجی واحد alu می‌باشد نیاز است. دستورهای lb و sb نیز از طریق سیگنال کنترلی که از Control Unit خارج می‌شود قابل اجرا هستند.